

## THE 80 NOTEBOOK

OCTOBER 1980

ISSUE #7

.....

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation. THE 80 NOTEBOOK is not affiliated with Radio Shack or Tandy Corporation in any way.

.....

## MACHINE LANGUAGE AND ASSEMBLY LANGUAGE:

## PROGRAMMING MADE AS SIMPLE AS BASIC

(continued from the September issue)

In last month's issue, we began our exploration of the world of machine and assembly language with a review of memory functions, the hardware architecture of the TRS-80, addressing and the instruction set our system will use. Now we will continue our study of this topic with a look at how to use our instructions to accomplish some tasks. The specific tasks we will examine involve making use of the I/O service routines in the BASIC ROM available to an assembly language application program.

First, at location address 16416 in memory, there is a DA field containing the address within the screen memory area of the cursor character on the display screen. The address value in the field may range from 15360 to 16383. Once the address value has been stored in the DA field, you will want to turn the cursor character on. This can be done by the following routine:

```
PUSH DE
PUSH IY
LD  A,14
CALL 9,51
POP  IY
POP  DE
```

To turn the cursor off at its current location, use the following routine:

```
PUSH DE
PUSH IY
LD  A,15
CALL 9,51
POP  IY
POP  DE
```

To display a character of text at the current cursor location on the screen, place the character of text in the A register and perform the following routine:

```

PUSH DE
PUSH IY
CALL 9,51
POP IY
POP DE

```

When an assembly language program wishes to obtain input data from the TRS-80 keyboard, it must scan the keyboard for a struck key, receive the representative character and then repeat that process for each character of text expected or until a termination character, such as a return key character, is received. As each character is received in the A register, it must be stored in an area of memory reserved in your program for keyboard input. If you are receiving a field of numeric data, you should remember that the numeric text as stored in your memory is in string form. If you wish to perform any arithmetic operations on that data, you will need to write an assembly language routine to convert the numeric string into an 8 or 16 bit machine language number format first. Even then you will still have to take signs and decimal points into consideration during any arithmetic manipulation of the converted numeric data. The following is an example of how to scan the keyboard until a character is received in the A register:

```

TAG AGN
PUSH DE
PUSH IY
CALL 9,43
OR A
JR 5,AGN
POP IY
POP DE

```

An assembly language program might also use data from or write data to cassette tape. Reading a record from a cassette is a four step operation. First you might define the cassette drive:

```

LD A,0
CALL 9,530

```

Then you must find the next sync byte on the tape:

```

CALL 9,662

```

Then you must read the record into memory one byte at a time. To read a 100 character record into an area in memory labeled RECA:

```

LD B,100
LDX HL,ARECA
TAG LBLA
PUSH BC
PUSH HL
CALL 9,565
POP HL
POP BC
LD M,A
INX HL
LD A,B

```

```

DEC  A
LD   B,A
JR   5,LBLA
DS   100,RECA
DA   RECA,ARECA

```

Finally, you must turn the cassette drive off:

```
CALL 9,504
```

Likewise, to write a record to the cassette tape involves four similar steps. First, you must define the cassette drive. Then, you must write a leader and sync byte:

```
CALL 9,647
```

Then, you must write your record to tape one byte at a time. To write a 100 byte record stored in a memory area labeled RECA to tape:

```

LD   B,100
LDX  HL,ARECA
TAG  LBLA
PUSH BC
PUSH HL
LD   A,M
CALL 9,612
POP  HL
POP  BC
LD   A,B
INX  HL
DEC  A
LD   B,A
JR   5,LBLA
DS   100,RECA
DA   RECA,ARECA

```

Finally, you must turn off the cassette drive.

If you have a stand-alone (not called by a user BASIC program) assembly language program, you should end your program by jumping to the BASIC command level ready state:

```
JP 9,6681
```

For those assembly language routines which are called by user written BASIC programs, the following conventions should be observed: To call an assembly language routine from BASIC, you must first identify the location of the routine and store it in a special area of memory at locations 16526 and 16527. For example, if your routine starts at location 32000, you convert this address into its 2 byte 16 bit format. In this form, byte 1 of the address would contain the value 208 and byte 2 the value 0. These 2 bytes must be stored in reverse order because they represent an address. This can be done by the following:

```
10 POKE 16526,0:POKE 16527,208
```

You then call the routine using the USR command. If a signed 16 bit value

is to be passed to your routine as an argument, it can be done through a numeric literal or field name argument expressed with the USR command. If no argument is to be passed, a dummy argument of zero is expressed with the USR command. The following is an example of a USR call with an argument contained in a field named N:

```
15 N=1295
20 A=USR(N)
```

If your routine is expecting an argument, the following instruction will retrieve the value and place it in the HL register pair:

```
CALL 9,2687
```

Remember, this value is a signed 16 bit value which means the possible values range from -32768 to +32767 because the first bit (leftmost, bit 1) represents a sign (a 1 bit represents a negative) rather than a numeric value position. At the end of your routine execution, if no argument value is being passed back to your BASIC program, you should end your routine with a RET instruction. If you are returning a signed 16 bit value (which would be placed in the A field of line 20 in our example), load the value into the HL register pair and execute a JP 9,2714 instruction at the end of your routine.

All assembly language programs and subroutines should begin by storing the current SP register value in a save area in your program and a new SP register value loaded based on a large enough stack area in your program to handle your program's needs. The saved SP register value must be re-loaded before you end your program or routine.

Now that we have a fundamental understanding of assembly language, we can discuss how to input and prepare a program or routine for use on your TRS-80. This process is divided into three parts: a program editor for inputting and modifying assembly language programs, a compiler for translating assembly language programs into machine language, and a relocating loader for loading a machine language program into a reserved section of memory and resolving its address usage to match the address available within its reserved section of memory.

Once an assembly language program has been inputted through the program editor, it is stored on cassette tape for future modification or compilation. Should you wish to modify an existing program, the editor will first read your program back into its buffer, accept your modifications and then rewrite your program back to tape. Here is the BASIC program that will serve as our editor:

```
5 INPUT "HOW MUCH STRING SPACE SHALL I CLEAR";C:CLEAR C
10 INPUT "ARE YOU 1) CREATING OR 2) MODIFYING A PROGRAM";A
15 INPUT "MAXIMUM NUMBER OF INSTRUCTIONS IN YOUR PROGRAM";B
20 DIM K$(B):N=0
25 FOR I=1 TO B:K$(I)=" ":NEXT I
30 IF A=2 THEN 200
35 PRINT "ENTER YOUR PROGRAM ONE INSTRUCTION AT A TIME"
40 IF N=B THEN PRINT "BUFFER FULL":GOTO 60
45 INPUT M$
50 IF M$="/" OR M$="*" THEN 60
55 N=N+1:K$(N)=M$:GOTO 40
```

```

60 PRINT "READY CASSETTE TO SAVE YOUR PROGRAM"
65 FOR I=1 TO N
75 PRINT #1,CHR$(34)+K$(I)+CHR$(34)
80 NEXT I:PRINT #1,CHR$(34)+"/"+CHR$(34)
85 INPUT "DO YOU WISH A PROGRAM LISTING";M$
90 IF M$.NE."YES" THEN END
95 FOR I=1 TO N:LPRINT I;K$(I):PRINT I;K$(I):NEXT I:END
200 PRINT "READY CASSETTE TO LOAD YOUR PROGRAM"
205 INPUT #1,M$
210 IF M$="/" THEN 220
215 N=N+1:K$(N)=M$:IF N=B THEN PRINT "BUFFER FULL":GOTO 220 ELSE 205
220 PRINT "L FOR LIST, I FOR INSERT, C FOR CHANGE, E TO FINISH"
225 INPUT "WHAT FUNCTION";M$:IF M$="E" THEN 60
230 IF M$="I" THEN 300
235 IF M$="C" THEN 400
240 IF M$.NE."L" THEN 225
245 INPUT "ENTER FROM, TO LINE #S";R,T
250 IF R.GT.T OR R.LT.1 OR R.GT.N OR T.GT.N THEN 245
255 FOR I=R TO T:PRINT I;K$(I):NEXT I:GOTO 220
300 INPUT "PLACE INSERTS AFTER LINE #";R:IF R.LT.1 OR R.GT.N THEN 300
305 INPUT "NUMBER OF INSERTED LINES";T:IF T.LT.1 THEN 305
310 IF N+T.GT.B THEN PRINT "NOT ENOUGH ROOM":GOTO 300
315 S=N:FOR I=1 TO N-R:K$(S+T)=K$(S):S=S-1:N=N+T
320 PRINT "ENTER INSERTED LINES ONE INSTRUCTION AT A TIME"
325 FOR I=1 TO T:INPUT M$:K$(R+I)=M$:NEXT I:GOTO 220
400 INPUT "LINE # BEING CHANGED";R:IF R.GT.N OR R.LT.1 THEN 400
405 PRINT "OLD LINE";K$(R)
410 INPUT "NEW LINE";M$:K$(R)=M$:GOTO 220

```

Note that when initially creating a program, the last line must be a "/" . Each line of instruction as entered or changed with this editor would consist of an operation code followed by a blank followed by up to two operands (if required) separated by only a comma. Naturally, the blank and comma would not be needed for an operation code requiring no operands.

If you wish to have comments in your program for documentation purposes, you may include them by placing them on separate lines starting with a ";" character. These comment lines will be ignored by the compiler.

When modifying a program, you may list portions of your program on the screen by specifying starting and ending line numbers, insert additional lines of instructions immediately following a specified line number or just replace a current line of instruction with a new one. As you can see, you can also obtain a listing of your program on your line printer.

When using this editor, you must have some idea how big your program will be in terms of lines of instructions. This information must be used to determine how much string space to clear since the editor's program buffer resides in BASIC's string space and must hold the whole program at one time. Normally, 20 to 30 characters per line is ample.

The compiler program will read your assembly language program tape and translate it into a machine language program which will be outputted to tape for further processing and execution by the relocating loader program. Optionally, a line printer listing of the compilation may be obtained. Here is the BASIC program that will serve as our assembly language compiler:

```

10 INPUT "HOW MUCH STRING SPACE SHALL I CLEAR";C: CLEAR C
15 INPUT "MAXIMUM NUMBER OF INSTRUCTIONS IN YOUR PROGRAM";B
20 INPUT "MAXIMUM # OF TAGS IN YOUR PROGRAM";A
25 INPUT "MAXIMUM # OF BYTES IN YOUR MACHINE LANGUAGE PROGRAM";F
27 F=F+10
30 DIM K$(B),J$(A),E%(A),H%(F),Z%(4)
35 G=(F/2)+5:DIM L%(G,2):N=0:P=0:Q=0:R=0:T=1:EC=0
40 INPUT "READY CASSETTE TO READ YOUR ASSEMBLY PROGRAM";YB$
45 INPUT #-1,M$:IF M$="/*" THEN 55
50 N=N+1:K$(N)=M$:IF N=B THEN PRINT "BUFFER FULL":GOTO 55 ELSE 45
55 FOR I=0 TO A:J$(I)=" ":E%(I)=0:NEXT I
60 FOR I=0 TO F:H%(I)=0:NEXT I
65 FOR I=0 TO G:FOR V=0 TO 2:L%(I,V)=0:NEXT V:NEXT I
70 INPUT "WANT A LINE PRINTER LISTING";YA$
75 PRINT "*LOC*NB*MACHINE CODE** *ASSEMBLY TEXT*"
80 IF YA$="YES" THEN LPRINT "*LOC*NB*MACHINE CODE** *ASSEMBLY TEXT*"
85 FOR I=1 TO N:Z%(1)=0:Z%(2)=0:Z%(3)=0:Z%(4)=0:T9=T:M$=K$(I):Z5=LEN(M$)
87 ER$=" ":CP$=" ":X1$=" ":X2$=" ":NB=0
90 IF LEFT$(M$,1)=";" THEN 90000
95 FOR Z6=1 TO Z5:IF MID$(M$,Z6,1)=" " THEN 105
100 NEXT Z6:OP$=M$:GOTO 200
105 OP$=MID$(M$,1,Z6-1):Z6=Z6+1:IF Z6.GT.Z5 THEN 200
110 FOR Z7=Z6 TO Z5:IF MID$(M$,Z7,1)="," THEN 120
115 NEXT Z7:X1$=MID$(M$,Z6):GOTO 200
120 X1$=MID$(M$,Z6,Z7-Z6):Z7=Z7+1:IF Z7.GT.Z5 THEN 200
125 X2$=MID$(M$,Z7)
200 IF OP$.NE."TAG" THEN 230
205 IF X1$=" " THEN 1225
207 LA$=X1$:GOSUB 210:GOTO 90000
210 IF P=0 THEN 225
215 FOR V=1 TO P:IF J$(V)=LA$ AND E%(V).GT.0 THEN ER$="DUPLICATE LABEL":RETURN
217 IF J$(V)=LA$ THEN 227
220 NEXT V
225 P=P+1:V=P:IF P.GT.A THEN ER$="TOO MANY LABELS":RETURN
227 J$(V)=LA$:E%(V)=T:RETURN
230 IF OP$.NE."ORG" THEN 245
235 V=VAL(X1$):IF T+V.LT.0 OR T+V.GT.F THEN ER$="INVALID ADDRESS":GOTO 90000
240 T=T+V:GOTO 90000
245 IF OP$.NE."DS" THEN 265
250 V=VAL(X1$):IF V.LT.1 OR V.GT.32767 OR T+V.GT.F THEN ER$="INVALID ALLOCATION":GOTO 90000
255 IF X2$.NE." " THEN LA$=X2$:GOSUB 210
260 IF ER$=" " THEN T=T+V
262 GOTO 90000
265 IF OP$.NE."DC" THEN 305
270 IF LEFT$(X1$,1).NE.CHR$(39) OR RIGHT$(X1$,1).NE.CHR$(39) THEN ER$="INVALID SYNTAX":GOTO 90000
275 W=LEN(X1$)-2:IF W.LT.1 OR T+W.GT.F THEN ER$="INVALID SIZE":GOTO 90000
280 LA$=MID$(X1$,2,W)
285 FOR V=1 TO W:IF V.LT.5 THEN Z%(V)=ASC(MID$(LA$,V,1)):NB=NB+1
290 H%(V)=ASC(MID$(LA$,V,1)):NEXT V
295 IF X2$.NE." " THEN LA$=X2$:GOSUB 210
300 IF ER$=" " THEN T=T+W
302 GOTO 90000
305 IF OP$.NE."DB" THEN 350

```

```

310 W=LEN(X18):IF W.NE.8 THEN 1225
315 V=128:ZA%=0:FOR Y=1 TO W
320 IF MID$(X1$,Y,1)="1" THEN ZA%=ZA%+V:GOTO 330
325 IF MID$(X1$,Y,1).NE."0" THEN ER$="INVALID BIT":GOTO 9000
330 V=V/2:NEXT Y:H%(T)=ZA%:Z%(1)=ZA%:NB=1
332 IF T+1.GT.F THEN ER$="END OF MEMORY":GOTO 9000
335 IF X2$.NE." " THEN LA$=X2$:GOSUB 210
340 IF ER$=" " THEN T=T+1
345 GOTO 9000
350 IF OP$.NE."DW" THEN 395
355 V=VAL(X1$):IF V.LT.0 OR V.GT. 65535 THEN 1225
360 ZA%=INT(V/265):ZB%=V-(ZA%*265)
370 IF T+2.GT.F THEN ER$="END OF MEMORY":GOTO 9000
375 NB=NB+1:Z%(NB)=ZB%:NB=NB+1:Z%(NB)=ZA%:H%(T)=ZB%:H%(T+1)=ZA%
380 IF X2$.NE." " THEN LA$=X2$:GOSUB 210
385 IF ER$=" " THEN T=T+2
390 GOTO 9000
395 IF OP$.NE."DN" THEN 420
400 V=VAL(X1$):IF V.LT.-128 OR V.GT.127 THEN 1225
405 IF V.LT.0 THEN V=256-ABS(V)
415 NB=1:Z%(1)=V:H%(T)=V:GOTO 332
420 IF OP$.NE."DA" THEN 475 ELSE Y1=T
425 IF VAL(X1$).GT.0 THEN 355 ELSE GOSUB 430:GOTO 465
430 IF P=0 THEN 445
435 FOR V=1 TO P:IF J$(V)=X1$ THEN 455
440 NEXT V
445 P=P+1:V=P:IF P.GT.A THEN ER$="TOO MANY LABELS":RETURN
450 J$(V)=X1$:E%(V)=0:Y5=V
455 Q=Q+1:IF Q.GT.G THEN ER$="TOO MANY RELOCATIONS":RETURN
460 L%(Q,1)=Y1:L%(Q,2)=V:RETURN
465 IF ER$.NE." " THEN 9000
470 V=E%(Y5):GOTO 360
475 IF OP$.NE."DM" THEN 490
480 V=VAL(X1$):IF V.LT.0 OR V.GT.255 THEN 1225
485 GOTO 415
490 IF OP$.NE."RLC" THEN 600
495 LA$=X1$:GOSUB 500:GOTO 575
500 V=-1:V1=-1:W=LEN(LA$):IF W.LT.2 THEN 530
505 IF LEFT$(LA$,2)="IX" THEN NB=NB+1:Z%(NB)=221:GOTO 520
510 IF LEFT$(LA$,2)="IY" THEN NB=NB+1:Z%(NB)=253:GOTO 520
515 ER$="INVALID OPERAND":RETURN
520 V=6:IF W.GT.3 THEN 525 ELSE RETURN
525 V1=VAL(MID$(LA$,4)):RETURN
530 IF LA$="M" THEN V=6:RETURN
535 IF LA$="A" THEN V=7:RETURN
540 IF LA$="B" THEN V=0:RETURN
545 IF LA$="C" THEN V=1:RETURN
550 IF LA$="D" THEN V=2:RETURN
555 IF LA$="E" THEN V=3:RETURN
560 IF LA$="H" THEN V=4:RETURN
565 IF LA$="L" THEN V=5:RETURN
570 GOTO 515
575 IF V=-1 THEN 9000
577 NB=NB+1:Z%(NB)=203:NB=NB+1
580 IF V1.GT.-1 THEN Z%(NB)=V1:NB=NB+1
585 Z%(NB)=V

```

```

590 IF T+NB.GT.F THEN ER$="END OF MEMORY":GOTO 9000
595 FOR W=1 TO NB:H%(T)=Z%(W):T=T+1:NEXT W:GOTO 9000
600 IF OP$.NE."RR" THEN 615
605 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN V=V+24
610 GOTO 575
615 IF OP$.NE."RRC" THEN 630
620 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN V=V+8
625 GOTO 575
630 IF OP$.NE."SBC" THEN 660
632 V=VAL(X1$):IF V.GT.0 AND V.LT.256 THEN NB=2:Z%(1)=222:Z%(2)=V:GOTO 590
635 LA$=X1$:GOSUB 500:W=152
640 IF V=-1 THEN 9000
645 V=V+W:NB=NB+1:Z%(NB)=V
650 IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
655 GOTO 590
660 IF OP$.NE."SBCX" THEN 705
665 LA$=X1$:GOSUB 670:GOTO 695
670 IF LA$="BC" THEN V=0:RETURN
675 IF LA$="DE" THEN V=16:RETURN
680 IF LA$="HL" THEN V=32:RETURN
685 IF LA$="SP" THEN V=48:RETURN
690 V=-1:ER$="INVALID OPERAND":RETURN
695 IF V=-1 THEN 9000
700 NB=NB+1:Z%(NB)=237:NB=NB+1:Z%(NB)=V+66:GOTO 590
705 IF OP$.NE."SCF" THEN 715
710 NB=1:Z%(1)=55:GOTO 590
715 IF OP$.NE."SET" THEN 735 ELSE WA=192
720 W=VAL(X1$):IF W.LT.1 OR W.GT.8 THEN 1225
725 LA$=X2$:GOSUB 500:IF V.GT.-1 THEN V=V+WA+(ABS(W-8)*8)
730 GOTO 575
735 IF OP$.NE."SLA" THEN 750
740 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN V=V+32
745 GOTO 575
750 IF OP$.NE."SRA" THEN 765
755 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN V=V+40
760 GOTO 575
765 IF OP$.NE."SRL" THEN 780
770 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN V=V+56
775 GOTO 575
780 IF OP$.NE."SUB" THEN 795
785 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=214:Z%(2)=W:GOTO 590
790 LA$=X1$:GOSUB 500:W=144:GOTO 640
795 IF OP$.NE."XOR" THEN 810
800 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=238:Z%(2)=W:GOTO 590
805 LA$=X1$:GOSUB 500:W=168:GOTO 640
810 IF OP$.NE."IN" THEN 825 ELSE V=219
815 W=VAL(X1$):IF W.LT.0 OR W.GT.255 THEN 1225
820 NB=2:Z%(1)=V:Z%(2)=W:GOTO 590
825 IF OP$.NE."LDD" THEN 835
830 NB=2:Z%(1)=237:Z%(2)=168:GOTO 590
835 IF OP$.NE."LDDR" THEN 845
840 NB=2:Z%(1)=237:Z%(2)=184:GOTO 590
845 IF OP$.NE."LDI" THEN 855
850 NB=2:Z%(1)=237:Z%(2)=160:GOTO 590
855 IF OP$.NE."LDIR" THEN 865
860 NB=2:Z%(1)=237:Z%(2)=176:GOTO 590
865 IF OP$.NE."NEG" THEN 875
870 NB=2:Z%(1)=237:Z%(2)=68:GOTO 590

```



```

875 IF OP$.NE."NOP" THEN 885
880 NB=1:Z%(1)=0:GOTO 590
885 IF OP$.NE."OR" THEN 900
890 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=246:Z%(2)=W:GOTO 590
895 LA$=X1$:GOSUB 500:W=176:GOTO 640
900 IF OP$.NE."OUT" THEN 910
905 V=211:GOTO 815
910 IF OP$.NE."POP" THEN 965
915 LA$=X1$:GOSUB 920:GOTO 955
920 IF LA$="AF" THEN V=241:RETURN
925 IF LA$="BC" THEN V=193:RETURN
930 IF LA$="DE" THEN V=209:RETURN
935 IF LA$="HL" THEN V=225:RETURN
940 IF LA$="IX" THEN V=225:NB=NB+1:Z%(NB)=221:RETURN
945 IF LA$="IY" THEN V=225:NB=NB+1:Z%(NB)=253:RETURN
950 V=-1:ER$="INVALID OPERAND":RETURN
955 IF V=-1 THEN 9000
960 NB=NB+1:Z%(NB)=V:GOTO 590
965 IF OP$.NE."PUSH" THEN 980
970 LA$=X1$:GOSUB 920:IF V.GT.-1 THEN V=V+4
975 GOTO 955
980 IF OP$.NE."RES" THEN 985 ELSE WA=128:GOTO 720
985 IF OP$.NE."RL" THEN 1000
990 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN V=V+16
995 GOTO 575
1000 IF OP$.NE."INC" THEN 1055
1005 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000
1010 IF V=0 THEN W=4
1015 IF V=1 THEN W=12
1020 IF V=2 THEN W=20
1025 IF V=3 THEN W=28
1030 IF V=4 THEN W=36
1035 IF V=5 THEN W=44
1040 IF V=6 THEN W=52
1042 IF V=7 THEN W=60
1045 NB=NB+1:Z%(NB)=W:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
1050 GOTO 590
1055 IF OP$.NE."INCX" THEN 1085
1057 IF X1$="SP" THEN W=51:GOTO 1080
1060 LA$=X1$:GOSUB 925:IF V=-1 THEN 9000
1065 IF V=193 THEN W=3
1070 IF V=209 THEN W=19
1075 IF V=225 THEN W=35
1080 NB=NB+1:Z%(NB)=W:GOTO 590
1085 IF OP$.NE."JP" THEN 1190
1090 W=VAL(X1$):IF W.LT.1 OR W.GT.9 THEN 1225
1095 IF W=9 AND X2$="M" THEN NB=1:Z%(NB)=223:GOTO 590
1100 IF W=9 AND X2$="IX" THEN NB=2:Z%(1)=221:Z%(2)=233:GOTO 590
1105 IF W=9 AND X2$="IY" THEN NB=2:Z%(1)=253:Z%(2)=233:GOTO 590
1110 IF W=1 THEN V=218
1115 IF W=2 THEN V=210
1120 IF W=3 THEN V=250
1125 IF W=4 THEN V=242
1130 IF W=5 THEN V=202
1135 IF W=6 THEN V=194
1140 IF W=7 THEN V=234
1145 IF W=8 THEN V=226

```

```

1150 IF W=9 THEN V=195
1155 NB=NB+1:Z%(NB)=V:V=VAL(X2$):Y1=T+1
1160 IF V.GT.0 AND V.LT.65536 THEN 1165 ELSE 1180
1165 ZA%=INT(V/256):V=V-(ZA%*256)
1175 NB=NB+1:Z%(NB)=V:NB=NB+1:Z%(NB)=ZA%:GOTO 590
1180 X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1185 V=E%(Y5):GOTO 1165
1190 IF OP$.NE."JR" THEN 1250
1195 W=VAL(X1$):IF W.LT.1 OR W.GT.9 THEN 1225
1200 IF W=1 THEN V=56:GOTO 1230
1205 IF W=9 THEN V=24:GOTO 1230
1210 IF W=2 THEN V=48:GOTO 1230
1215 IF W=6 THEN V=32:GOTO 1230
1220 IF W=5 THEN V=40:GOTO 1230
1225 ER$="INVALID OPERAND":GOTO 9000
1230 W=VAL(X2$):IF W.LT.-128 OR W.GT.127 THEN 1225
1235 IF W.LT.0 THEN W=256-ABS(W)
1245 NB=NB+1:Z%(NB)=V:NB=NB+1:Z%(NB)=W:GOTO 590
1250 IF OP$.NE."LD" THEN 1425 ELSE Y1=T+1
1255 IF X1$="@BC" AND X2$="A" THEN NB=1:Z%(1)=2:GOTO 590
1260 IF X1$="@DE" AND X2$="A" THEN NB=1:Z%(1)=18:GOTO 590
1265 V=VAL(X1$):IF V.GT.0 AND V.LT.65536 AND X2$="A" THEN NB=1:Z%(1)=50:GOTO 1165
1270 LA$=X1$:GOSUB 500:IF V.GT.-1 THEN 1282
1272 IF X2$.NE."A" THEN 9000
1275 ER$=" ":GOSUB 430:IF ER$.NE." " THEN 9000
1280 V=E%(Y5):NB=1:Z%(1)=50:GOTO 1165
1282 Y2=V:Y4=V1
1285 IF X1$="A" AND X2$="@BC" THEN NB=1:Z%(1)=10:GOTO 590
1290 IF X1$="A" AND X2$="@DE" THEN NB=1:Z%(1)=26:GOTO 590
1295 V=VAL(X2$):IF X1$="A" AND V.GT.255 AND V.LT.65536 THEN NB=1:Z%(1)=58:GOTO 1165
1296 V=0:W=0
1297 W=VAL(X2$):IF W.GT.0 AND W.LT.256 THEN 1320
1300 LA$=X2$:GOSUB 500:IF V.GT.-1 THEN 1320
1305 IF X1$.NE."A" THEN 9000
1310 ER$=" ":GOSUB 430:IF ER$.NE." " THEN 9000
1315 V=E%(Y5):NB=1:Z%(1)=58:GOTO 1165
1320 Y3=V:ON Y2+1 GOTO 1325, 1335, 1345, 1355, 1365, 1375, 1385, 1415
1325 IF W.GT.0 THEN NB=2:Z%(1)=6:Z%(2)=W:GOTO 590
1330 NB=NB+1:Z%(NB)=64+Y3:GOTO 650
1335 IF W.GT.0 THEN NB=2:Z%(1)=14:Z%(2)=W:GOTO 590
1340 NB=NB+1:Z%(NB)=72+Y3:GOTO 650
1345 IF W.GT.0 THEN NB=2:Z%(1)=22:Z%(2)=W:GOTO 590
1350 NB=NB+1:Z%(NB)=80+Y3:GOTO 650
1355 IF W.GT.0 THEN NB=2:Z%(1)=30:Z%(2)=W:GOTO 590
1360 NB=NB+1:Z%(NB)=88+Y3:GOTO 650
1365 IF W.GT.0 THEN NB=2:Z%(1)=38:Z%(2)=W:GOTO 590
1370 NB=NB+1:Z%(NB)=96+Y3:GOTO 650
1375 IF W.GT.0 THEN NB=2:Z%(1)=46:Z%(2)=W:GOTO 590
1380 NB=NB+1:Z%(NB)=104+Y3:GOTO 650
1385 IF Y4.GT.-1 THEN 1405
1390 IF W.GT.0 THEN NB=2:Z%(1)=54:Z%(2)=W:GOTO 590
1395 IF Y3=6 THEN 1225
1400 NB=1:Z%(1)=112+Y3:GOTO 650
1405 IF W.GT.0 THEN NB=4:Z%(2)=54:Z%(3)=Y4:Z%(4)=W:GOTO 590

```

```

1410 NB=3:Z%(2)=112+Y3:Z%(3)=Y4:GOTO 590
1415 IF W.GT.0 THEN NB=2:Z%(1)=62:Z%(2)=W:GOTO 590
1420 NB=NB+1:Z%(NB)=120+Y3:GOTO 650
1425 IF OP$.NE."LDX" THEN 1655
1430 W=VAL(X2$):V=VAL(X1$)
1435 IF V.GT.0 AND V.LT.65536 THEN 1440 ELSE 1475
1440 IF X2$="BC" THEN NB=2:Z%(1)=237:Z%(2)=67:GOTO 1165
1445 IF X2$="DE" THEN NB=2:Z%(1)=237:Z%(2)=83:GOTO 1165
1450 IF X2$="HL" THEN NB=1:Z%(1)=34:GOTO 1165
1455 IF X2$="IX" THEN NB=2:Z%(1)=221:Z%(2)=34:GOTO 1165
1460 IF X2$="IY" THEN NB=2:Z%(1)=253:Z%(2)=34:GOTO 1165
1465 IF X2$="SP" THEN NB=2:Z%(1)=237:Z%(2)=115:GOTO 1165
1470 GOTO 1225
1475 IF X1$.NE."BC" THEN 1500
1480 IF W.GT.0 AND W.LT.65536 THEN NB=2:Z%(1)=237:Z%(2)=75:V=W:GOTO 1165
1485 IF LEFT$(X2$,1)="/" THEN V=VAL(MID$(X2$,2)):NB=1:Z%(1)=1:GOTO 1165
1490 Y1=T+2:X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1495 V=E%(Y5):NB=2:Z%(1)=237:Z%(2)=75:GOTO 1165
1500 IF X1$.NE."DE" THEN 1525
1505 IF W.GT.0 AND W.LT.65536 THEN NB=2:Z%(1)=237:Z%(2)=91:V=W:GOTO 1165
1510 IF LEFT$(X2$,1)="/" THEN V=VAL(MID$(X2$,2)):NB=1:Z%(1)=17:GOTO 1165
1515 Y1=T+2:X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1520 V=E%(Y5):NB=2:Z%(1)=237:Z%(2)=91:GOTO 1165
1525 IF X1$.NE."HL" THEN 1550
1530 IF W.GT.0 AND W.LT.65536 THEN NB=1:Z%(1)=42:V=W:GOTO 1165
1535 IF LEFT$(X2$,1)="/" THEN V=VAL(MID$(X2$,2)):NB=1:Z%(1)=33:GOTO 1165
1540 Y1=T+2:X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1545 V=E%(Y5):NB=1:Z%(1)=42:GOTO 1165
1550 IF X1$.NE."IX" THEN 1575
1555 IF W.GT.0 AND W.LT.65536 THEN NB=2:Z%(1)=221:Z%(2)=42:V=W:GOTO 1165
1560 IF LEFT$(X2$,1)="/" THEN V=VAL(MID$(X2$,2)):NB=2:Z%(1)=221:Z%(2)=33:GO
    TO 1165
1565 Y1=T+2:X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1570 V=E%(Y5):NB=2:Z%(1)=221:Z%(2)=42:GOTO 1165
1575 IF X1$.NE."IY" THEN 1600
1580 IF W.GT.0 AND W.LT.65536 THEN NB=2:Z%(1)=253:Z%(2)=42:V=W:GOTO 1165
1585 IF LEFT$(X2$,1)="/" THEN NB=2:V=VAL(MID$(X2$,2)):Z%(1)=253:Z%(2)=33:GO
    TO 1165
1590 Y1=T+2:X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1595 V=E%(Y5):NB=2:Z%(1)=253:Z%(2)=42:GOTO 1165
1600 IF X1$.NE."SP" THEN 1640
1605 IF X2$="HL" THEN NB=1:Z%(1)=249:GOTO 590
1610 IF X2$="IX" THEN NB=2:Z%(1)=221:Z%(2)=249:GOTO 590
1615 IF X2$="IY" THEN NB=2:Z%(1)=253:Z%(2)=249:GOTO 590
1620 IF W.GT.0 AND W.LT.65536 THEN NB=2:Z%(1)=237:Z%(2)=123:V=W:GOTO 1165
1625 IF LEFT$(X2$,1)="/" THEN NB=1:V=VAL(MID$(X2$,2)):Z%(1)=49:GOTO 1165
1630 Y1=T+2:X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1635 V=E%(Y5):NB=2:Z%(1)=237:Z%(2)=123:GOTO 1165
1640 Y1=T+2:IF X2$="HL" THEN Y1=T+1
1645 GOSUB 430:IF ER$.NE." " THEN 9000
1650 V=E%(Y5):GOTO 1440
1655 IF OP$.NE."CPIR" THEN 1665
1660 NB=2:Z%(1)=237:Z%(2)=177:GOTO 590
1665 IF OP$.NE."CPL" THEN 1675
1670 NB=1:Z%(1)=47:GOTO 590
1675 IF OP$.NE."DEC" THEN 1735
1680 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000

```

```

1685 IF V=0 THEN W=5
1690 IF V=1 THEN W=13
1695 IF V=2 THEN W=21
1700 IF V=3 THEN W=29
1705 IF V=4 THEN W=37
1710 IF V=5 THEN W=45
1715 IF V=6 THEN W=53
1720 IF V=7 THEN W=61
1725 NB=NB+1:Z%(NB)=W:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
1730 GOTO 590
1735 IF OP$.NE."DECX" THEN 1770
1740 IF X1$="SP" THEN W=59:GOTO 1765
1745 LA$=X1$:GOSUB 925:IF V=-1 THEN 9000
1750 IF V=193 THEN W=11
1755 IF V=209 THEN W=27
1760 IF V=225 THEN W=43
1765 NB=NB+1:Z%(NB)=W:GOTO 590
1770 IF OP$.NE."DJNZ" THEN 1785
1775 W=VAL(X1$):IF W.LT.-128 OR W.GT.127 THEN 1225
1777 IF W.LT.0 THEN W=256-ABS(W)
1780 NB=2:Z%(1)=16:Z%(2)=W:GOTO 590
1785 IF OP$.NE."EXS" THEN 1810
1790 IF X1$="HL" THEN NB=1:Z%(1)=227:GOTO 590
1795 IF X1$="IX" THEN NB=2:Z%(1)=221:Z%(2)=227:GOTO 590
1800 IF X1$="IY" THEN NB=2:Z%(1)=253:Z%(2)=227:GOTO 590
1805 GOTO 1225
1810 IF OP$.NE."EXD" THEN 1820
1815 NB=1:Z%(1)=235:GOTO 590
1820 IF OP$.NE."EXA" THEN 1830
1825 NB=1:Z%(1)=8:GOTO 590
1830 IF OP$.NE."EXX" THEN 1840
1835 NB=1:Z%(1)=217:GOTO 590
1840 IF OP$.NE."BIT" THEN 1860
1845 W=VAL(X1$):IF W.LT.1 OR W.GT.8 THEN 1225
1850 LA$=X2$:GOSUB 500:IF V.GT.-1 THEN V=V+64+(ABS(W-8)*8)
1855 GOTO 575
1860 IF OP$.NE."CALL" THEN 1930 ELSE Y1=T+1
1862 W=VAL(X1$):IF W.LT.1 OR W.GT.9 THEN 1225
1865 V=VAL(X2$):IF V.GT.0 AND V.LT.65536 THEN 1880
1870 X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1875 V=E%(Y5)
1880 IF W=1 THEN Y2=220
1885 IF W=2 THEN Y2=212
1890 IF W=3 THEN Y2=252
1895 IF W=4 THEN Y2=244
1900 IF W=5 THEN Y2=204
1905 IF W=6 THEN Y2=196
1910 IF W=7 THEN Y2=236
1915 IF W=8 THEN Y2=228
1920 IF W=9 THEN Y2=205
1925 NB=NB+1:Z%(NB)=Y2:GOTO 1165
1930 IF OP$.NE."CCF" THEN 1940
1935 NB=1:Z%(1)=63:GOTO 590
1940 IF OP$.NE."CP" THEN 1965
1945 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=254:Z%(2)=W:GOTO 590
1950 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000
1955 NB=NB+1:Z%(NB)=184+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
1960 GOTO 590

```

```

1965 IF OP$.NE."CPD" THEN 1975
1970 NB=2:Z%(1)=237:Z%(2)=169:GOTO 590
1975 IF OP$.NE."CPDR" THEN 1985
1980 NB=2:Z%(1)=237:Z%(2)=185:GOTO 590
1985 IF OP$.NE."CPI" THEN 1995
1990 NB=2:Z%(1)=237:Z%(2)=161:GOTO 590
1995 IF OP$.NE."ADC" THEN 2020
2000 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=206:Z%(2)=W:GOTO 590
2005 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000
2010 NB=NB+1:Z%(NB)=136+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
2015 GOTO 590
2020 IF OP$.NE."ADCX" THEN 2055
2025 IF X1$="BC" THEN W=74:GOTO 2050
2030 IF X1$="DE" THEN W=90:GOTO 2050
2035 IF X1$="HL" THEN W=106:GOTO 2050
2040 IF X1$="SP" THEN W=122:GOTO 2050
2045 GOTO 1225
2050 NB=2:Z%(1)=237:Z%(2)=W:GOTO 590
2055 IF OP$.NE."ADD" THEN 2080
2060 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=198:Z%(2)=W:GOTO 590
2065 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000
2070 NB=NB+1:Z%(NB)=128+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
2075 GOTO 590
2080 IF OP$.NE."ADDX" THEN 2135
2085 IF X1$.NE."HL" THEN 2115
2087 NB=NB+1
2090 IF X2$="BC" THEN Z%(NB)=9:GOTO 590
2095 IF X2$="DE" THEN Z%(NB)=25:GOTO 590
2100 IF X2$="HL" THEN Z%(NB)=41:GOTO 590
2105 IF X2$="SP" THEN Z%(NB)=57:GOTO 590
2110 GOTO 1225
2115 IF X1$.NE."IX" THEN 2125
2120 NB=NB+1:Z%(NB)=221:GOTO 2087
2125 IF X1$.NE."IY" THEN 1225
2130 NB=NB+1:Z%(NB)=253:GOTO 2087
2135 IF OP$.NE."AND" THEN ER$="INVALID OP CODE":GOTO 9000
2140 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=230:Z%(2)=W:GOTO 590
2145 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000
2150 NB=NB+1:Z%(NB)=160+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
2155 GOTO 590
9000 PRINT T9;NB;Z%(1);Z%(2);Z%(3);Z%(4);" ";K$(I)
9005 IF YA$="YES" THEN LPRINT T9;NB;Z%(1);Z%(2);Z%(3);Z%(4);" ";K$(I)
9010 IF ER$.NE." " THEN PRINT ER$:PRINT:EC=EC+1
9015 IF ER$.NE." " AND YA$="YES" THEN LPRINT ER$:LPRINT" "
9020 NEXT I:IF P=0 THEN 9050
9025 FOR I=1 TO P:IF E%(I)=0 THEN 9030 ELSE 9040
9030 PRINT "LABEL UNKNOWN-";J$(I):EC=EC+1
9035 IF YA$="YES" THEN LPRINT "LABEL UNKNOWN-";J$(I)
9040 NEXT I:FOR V=0 TO Q:Z7=L%(V,2):L%(V,2)=E%(Z7):NEXT V
9050 IF EC.GT.0 THEN PRINT "ERRORS=";EC:END
9055 INPUT "READY CASSETTE TO WRITE MACHINE LANGUAGE";YB$
9060 PRINT #-1,T;Q:I=0
9065 PRINT #-1,"T",H%(I),H%(I+1),H%(I+2),H%(I+3),H%(I+4),H%(I+5),H%(I+6),H%(I+7),H%(I+8),H%(I+9)
9070 I=I+10:IF I.LE.T THEN 9065
9075 IF Q=0 THEN 9095
9080 I=1
9085 PRINT #-1,"Q",L%(I,1),L%(I,2),L%(I+1,1),L%(I+1,2),L%(I+2,1),L%(I+2,2),

```

```

L%(I+3,1),L%(I+3,2),L%(I+4,1),L%(I+4,2)
9090 I=I+5:IF I.LE.Q THEN 9085
9095 PRINT #-1,"E",,,,,,,,,,,,:END

```

The amount of string space you must clear for a compilation relates to the compiler tables. In creating a machine language program from your assembly language program, the compiler must build and maintain several tables. The K\$ table holds your assembly language programs. The J\$ and E\$ tables hold the tags and field names and their location addresses within your program. The H\$ table holds your machine language program. The L\$ table holds the location of an address in your program requiring relocation when your program is loaded for execution, and the relative address of the tag or field name at the location requiring the program's load address to be added to it and placed in your machine code. The J\$, E\$ and L\$ tables are maintained by two compiler subroutines starting at line 430 and at line 210. If no errors are found during the compilation, your machine language program will be written to tape in a format compatible with the relocating loader program. During the compilation, the program will ask if a listing of the compilation will be required on the line printer. As you can see, each assembly language instruction is covered in a separate section of the compiler program. You can study the machine language coding of any assembly language instruction by examining the appropriate section. In this version of the compiler, the DJNZ and JR instructions only allow an immediate value operand as the displacement value from -128 to +127 in jumping from these instructions.

Here is the relocating loader program you will need to load and execute your machine language program.

```

5 INPUT "NUMBER OF RELOCATION ADDRESSES";K:IF K.LT.1 OR K.GT.32768 THEN 5
7 DIM L%(K,2)
10 INPUT "LOAD ADDRESS";A:IF A.LT.16384 OR A.GT.65535 THEN 10
20 INPUT "MACHINE LANGUAGE PROGRAM AREA SIZE";B:IF B.LT.0 OR K*2.GT.B OR B.
   GT.32767 OR A+B.GT.65535 THEN 20
30 INPUT "READY CASSETTE TO READ MACHINE LANGUAGE PROGRAM";C$
40 INPUT #-1,T,Q:I=0:AX=A:J=1:IF T.GT.B THEN PRINT "TEXT SIZE ERROR":END
42 IF Q.GT.K THEN PRINT "RELOCATION TABLE SIZE ERROR":END
50 INPUT #-1,C$,H1,H2,H3,H4,H5,H6,H7,H8,H9,HA
55 IF C$.NE."T" THEN PRINT "LOAD TEXT ERROR":END
60 POKE A,H1:POKE A+1,H2:POKE A+2,H3:POKE A+3,H4
70 POKE A+4,H5:POKE A+5,H6:POKE A+6,H7:POKE A+7,H8
80 POKE A+8,H9:POKE A+9,HA:I=I+1:IF I.LE.T THEN 50
90 INPUT #-1,C$,H1,H2,H3,H4,H5,H6,H7,H8,H9,HA
100 IF C$="E" THEN 160
110 IF C$.NE."Q" THEN PRINT "RELOCATION TABLE ERROR":END
120 L%(J,1)=H1:L%(J,2)=H2:L%(J+1,1)=H3
130 L%(J+1,2)=H4:L%(J+2,1)=H5:L%(J+2,2)=H6
140 L%(J+3,1)=H7:L%(J+3,2)=H8:L%(J+4,1)=H9
150 L%(J+4,2)=HA:J=J+5:IF J.LE.Q THEN 90
160 IF J=1 THEN 210
170 FOR I=1 TO J:IF L%(I,1)=0 THEN 200
180 Z%=A+L%(I,2):H1=L%(I,1):W%=INT(Z%/256):Z%=Z%-(W%*256)
190 POKE H1,Z%:POKE H1+1,W%
200 NEXT I
210 INPUT "SHOULD I EXECUTE YOUR PROGRAM";C$
220 IF C$.NE."YES" THEN END
230 Z%=INT(A/256):W%=A-(Z%*256)

```

240 POKE 16526,W%:POKE 16527,Z%  
250 X=USR(0):END

Note that the load address, machine program size, and relocation address table size must match or exceed the capacities of those parameters as defined during the compilation of your program. The loader first processes the text records of machine code read from the tape and then reads all the address relocation records for further processing. Once the address relocation has been accomplished, the program may be executed. If the program is a subroutine to be called from one of your BASIC programs, the loader ends so that you may load and execute your BASIC program.

I think you will agree that this system for using assembly language and machine language programs is both simple and beneficial. Now all you need is some imagination and you can develop many interesting and fast running programs with features beyond BASIC's capabilities.

.....

#### CYBERMATE AND THE 80 NOTEBOOK GOING OUT OF BUSINESS

Due to overwhelming competition (trying to get a piece of the market) from the over 3000 small TRS-80 products firms, sources of cost effective advertising (since the CIE S-80 Bulletin went out of business in August), very poor circulation (presently about 350) and poor sales this past summer, we are forced to go out of business. We appreciate your interest in our company's products and publications and wish to leave you on a friendly basis.

Due to our current poor financial condition, we are offering the following: for those of you with one year subscriptions, we are including a copy of our 41 program listing package with this last issue of The 80 Notebook. Our package is an ideal substitute for the remainder of a one year subscription and it represents more material than was planned for the remainder of the first subscription year. If you already own one of our 41 program packages, keep this second copy and you may request any two programs on cassette from our product line of 41 programs.

Our two year subscribers can request 5 programs on cassette as a substitute to their second year of The 80 Notebook.

Our three year subscribers can request 10 programs on cassette as a substitute to their second and third years of The 80 Notebook.

We hope this will be satisfactory as we have done our best to be fair with the resources we have left.

GOOD-BYE AND GOOD LUCK!

